IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 July 2024, accepted 6 September 2024, date of publication 16 September 2024, date of current version 31 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3461773

RESEARCH ARTICLE

Faster ISNet for Background Bias Mitigation on Deep Neural Networks

PEDRO R. A. S. BASSI[1,2], SERGIO DECHERCHI[3], AND ANDREA CAVALLI[1,4,5]

[1]Alma Mater Studiorum, University of Bologna, 40126 Bologna, Italy
[2]Center for Biomolecular Nanotechnologies, Istituto Italiano di Tecnologia, 73010 Arnesano, Italy
[3]Data Science and Computation Facility, Istituto Italiano di Tecnologia, 16152 Genoa, Italy
[4]Centre Européen de Calcul Atomique et Moléculaire (CECAM), Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland
[5]Department of Computational and Chemical Biology, Istituto Italiano di Tecnologia, 16152 Genoa, Italy

Corresponding author: Pedro R. A. S. Bassi (pedro.salvadorbassi2@unibo.it)

This work was supported in part by the Center for Biomolecular Nanotechnologies, Istituto Italiano di Tecnologia, Arnesano, Italy; and in part by the Fondazione Istituto Italiano di Tecnologia, Genoa, Italy.

ABSTRACT Bias or spurious correlations in image backgrounds can impact neural networks, causing shortcut learning (*Clever Hans* Effect) and hampering generalization to real-world data. ISNet, a recently introduced architecture, proposed the optimization of Layer-Wise Relevance Propagation (LRP, an explanation technique) heatmaps, to mitigate the influence of backgrounds on deep classifiers. However, ISNet's training time scales linearly with the number of classes in an application. Here, we propose reformulated architectures, dubbed Faster ISNets, whose training time becomes independent from this number. Additionally, we introduce a concise and model-agnostic LRP implementation, LRP-Flex, which can readily explain arbitrary DNN architectures, or convert them into Faster ISNets. We challenge the proposed architectures using synthetic background bias, and COVID-19 detection in chest X-rays, an application that commonly presents background bias. The networks hindered background attention and shortcut learning, surpassing multiple state-of-the-art models on out-of-distribution test datasets. Representing a potentially massive training speed improvement over ISNet, the proposed architectures introduce LRP optimization into a gamut of applications that the original ISNet model cannot feasibly handle. Code for the Faster ISNet and LRP-Flex is available at https://github.com/PedroRASB/FasterISNet.

INDEX TERMS Shortcut learning, layer-wise relevance propagation, COVID-19 detection, explainable artificial intelligence, background bias, ISNet.

I. INTRODUCTION

Deep neural networks (DNNs) can achieve high or even super-human test accuracy in some computer vision tasks. However, sometimes such performances significantly drop when models are deployed in the real-world. Shortcut learning, or the *Clever Hans* effect [1], is a possible cause for this generalization gap [1]. Shortcuts, or spurious correlations, are image features that correlate with the classification labels in a training dataset, but these features are not reliably present in images drawn from data distributions other than the one that originated the training data. Shortcut learning characterizes neural networks learning decision rules that erroneously take advantage of spurious correlations [1]. As a result, these models perform well on standard test datasets, which are independent and identically distributed (i.i.d.) with respect to the training data. However, they have impaired generalization skills in out-of-distribution (o.o.d.) data, which commonly characterizes real-world applications. For this reason, shortcut learning is an obstacle for the wide and reliable utilization of DNNs in critical scenarios, such as medical or security-related tasks.

Background bias are shortcuts in images' backgrounds. COVID-19 detection in chest X-rays is a recent example of a critical application where background bias, and consequent shortcut learning, is common [2]. For a long time, most

The associate editor coordinating the review of this manuscript and approving it for publication was Rajeeb Dey.

large open access COVID-19 X-ray datasets contained no (or few) images displaying other diseases or healthy subjects, considering the same sources as the COVID-19 X-rays [3], [4], [5]. Therefore, many studies employed mixed datasets to train DNNs to classify COVID-19, healthy and other diseases [6]. Here, the term mixed datasets designates databases where images representing diverse classes come from distinct sources (e.g., different hospitals and cities). These sources may introduce different background characteristics in the X-rays images. As mixed datasets associate distinct sources to diverse classes, these characteristics become background bias and prompt shortcut learning. Some DNNs classifying COVID-19 performed exceedingly well on standard i.i.d. test datasets (e.g., with accuracies close to 100%), but latter studies demonstrated that such results were boosted and affected by shortcut learning. Accordingly, the networks performance dramatically dropped when these models were evaluated with X-rays collected from hospitals that did not contribute to their training datasets (o.o.d. testing) [2], [3], [7], [8].

Layer-Wise Relevance Propagation (LRP) [9] is a technique designed to explain DNN per-sample decisions. LRP, for one sample, whose label is predicted, creates a heatmap, namely a figure composed of a back-propagated quantity called relevance. Its magnitude indicates how much each part of the input image influenced the DNN's outputs. In a previous study, we presented a DNN architecture named ISNet [3]. It introduces a new use for LRP, by directly optimizing LRP heatmaps to improve a deep classifier's behavior. The model produces differentiable heatmaps during training, and feeds them to a heatmap loss [3]. The loss function uses segmentation masks to identify and penalize background attention in the maps. By minimizing a linear combination of the heatmap loss and a classification loss, the ISNet learns decision rules that ignore background bias. The acronym ISNet stands for ''Implicit Segmentation Neural Network'', referring to the network's ability to ignore images' backgrounds, as if they had been segmented out [3]. However, the network is a classifier and not a segmenter: it does not create segmentation masks. The ISNet's optimization procedure is called background relevance minimization, and it can be regarded as an explanation-based spatial attention mechanism. Notably, the ISNet does not need LRP heatmaps nor segmentation masks for inference, ensuring that the run-time model has no extra computation cost in relation to a standard classifier. Thus, it can be efficiently deployed in portable or embedded devices.

The ISNet was tested on synthetic background bias (inserted into natural images), and on standard mixed tuberculosis and COVID-19 X-ray datasets. By disregarding the background in its decisions, the ISNet hindered shortcut learning and improved generalization. Thus, performance on o.o.d. test datasets surpassed the multiple state-of-the-art DNNs it was compared to. Besides empirically comparing the ISNet to several alternative methodologies, the study presenting the ISNet explained in detail the algorithmic reasons for its superior performance [3]. Besides the ISNet, the usual segmentation-classification pipeline was the only other approach whose decisions were consistently not influenced by background bias. However, this model was less accurate, heavier and much slower in inference (needing to first run a deep segmenter, then classify the image with erased background). The ISNet's main drawback is that its training time linearly increases with the number of categories in the classification task [3]. Accordingly, ISNet's training computational cost becomes unfeasible for problems with many classes.

Our main contributions are:

1) To propose three deep classifier architectures, collectively named Faster ISNet. By reformulating the ISNet learning procedure, training time becomes independent of the number of classes. Hence, the models are drastically faster than the original ISNet when the number of classes is significant. Accordingly, this study feasibly introduces LRP optimization into a new gamut of applications.
2) To create LRP-Flex, a LRP implementation based on a reformulation of LRP's rules. Its main advantages over most LRP implementations (including ISNet's [3]) is conciseness and model agnosticism, not requiring architecture-specific code. Thus, LRP-Flex is a practical tool to explain arbitrary DNNs. The original ISNet was only implemented for three backbone architectures, but LRP-Flex can promptly convert any ReLU-based DNN into an ISNet or Faster ISNet.
3) To provide a set of techniques to better stabilize and accelerate ISNet training:
    a) We modified the ISNet loss, improving its convergence.
    b) We suggested a new heuristic to accelerate ISNet's hyper-parameter search.
    c) We introduced LRP Deep Supervision, a technique to improve the ISNet convergence and background bias robustness.

Here, we experiment with DenseNet [10] and the popular ResNet [11] backbones. To quantitatively measure the Faster ISNet's background bias robustness, we challenge it with synthetic background bias, which we inserted in the MNIST [12] and Stanford Dogs [13] datasets. Additionally, we train the DNNs for COVID-19 detection on chest X-rays, using the mixed dataset presented in the original ISNet study [3]. We evaluate the DNNs with X-rays from hospitals and cities that did not contribute to the training data, assessing whether the Faster ISNet hinders background bias attention and the consequent shortcut learning, thus improving o.o.d. generalization. We compare our model to the original ISNet and the multiple state-of-the-art classifier architectures used as benchmark in the original ISNet study [3]. Code for the Faster ISNet and LRP-Flex is available at https://github.com/PedroRASB/FasterISNet.

II. METHODS

A. LAYER-WISE RELEVANCE PROPAGATION

DNNs are complex non-linear models, with possibly millions of parameters. Therefore, understanding the reasons for a DNN's decision is challenging. However, this knowledge is fundamental for critical and security related applications (such as AI-assisted diagnosis), where trustworthiness is key. Layer-Wise Relevance Propagation [9] (LRP) is an explanation technique, i.e., a method to elucidate the reasons for a DNN output. LRP creates heatmaps, figures that indicate how each element (e.g., pixel) in the DNN input influenced the model's output. LRP backpropagates a quantity called relevance, from a chosen DNN output (logit), up to the network's input. The final relevance values constitute the heatmap, which bears the same shape as the input image. Positive relevance shows input regions that were responsible for increasing the chosen logit, i.e. they represent positive evidence for the class associated with the logit. Conversely, negative relevance indicates input areas that contributed to reducing the logit value, constituting negative evidence. Moreover, the relevance's absolute value informs how important the input region was for the DNN decision. The LRP propagation can start from any DNN logit, and the resulting heatmap will reveal positive and negative evidence for the class associated with such logit.

LRP uses semi-conservative rules to propagate the relevance layer-by-layer. In other words, the rules are designed to produce minimal destruction or creation of relevance, mostly redistributing it [14]. This property ensures a strong relationship between the heatmap elements and the DNN output [14]. For simplicity, here we explain how LRP-ε and LRP-z$^+$ propagate relevance through a fully-connected layer L with ReLU (Rectified Linear Unit) activation. However, the same equations are valid for convolutional layers followed by ReLU, as their pre-activation outputs are also linear combinations of their inputs. Moreover, the equations can also apply to batch normalization, dropout, pooling, and other common layers, by expressing them as equivalent fully-connected or convolutional layers, or fusing them with adjacent dense layers or convolutions [3], [14].

Equation (1) shows a fully-connected layer's outputs, z_k^L (L is a layer identification superscript), before the ReLU function:

$$z_k^L = \sum_j w_{jk}^L a_j^L \tag{1}$$

$$R_j^L = \sum_k \frac{w_{jk}^L a_j^L}{z_k^L + \text{sign}(z_k^L)\epsilon} R_k^{L+1} \tag{2}$$

$$R_k^{\text{Lmax}+1} = \begin{cases} z_c^{\text{Lmax}} & \text{if } k = c \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

w_{jk}^L represents the weight connecting the layer's input j to its output k, w_{0k}^L is the output k bias parameter (consider $a_0^L = 1$), and a_j^L is the layer's j-th input. Equation (2) represents LRP-ε [9]. It redistributes the relevance at the layer L output, R_k^{L+1} (i.e., the relevance referent to the input of layer L+1, $a_k^{L+1} = \text{ReLU}(z_k^L)$), to the layer's inputs (a_j^L), producing their respective relevances, R_j^L. Here, the sign(·) function returns 1 for positive or 0 arguments, and −1 for negative ones. The ϵ hyper-parameter is a small positive constant, used to avoid division by zero, improve numerical stability, and denoise heatmaps [3]. If $\epsilon = 0$, eq. (2) represents LRP-0 instead of LRP-ε. The quotient in eq. (2) shows that LRP redistributes relevance according to how much each input element, a_j^L, contributed to each layer output, z_k^L. Equation (3) shows the initial relevance values when explaining the logit relative to class c (z_c^{Lmax}, where Lmax indicates the last DNN layer).

The LRP-z$^+$ rule [15] (eq. (4)) is similar to LRP-ε, but it only propagates positive relevance:

$$R_j^L = \sum_k \frac{(w_{jk}^L a_j^L)^+}{\sum_j (w_{jk}^L a_j^L)^+ + \epsilon} R_k^{L+1} \tag{4}$$

$$R_k^{\text{Lmax}+1} = \begin{cases} \sum_j (w_{jc}^{\text{Lmax}} a_j^{\text{Lmax}})^+ & \text{if } k = c \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Thus, it ignores the negative evidence for a DNN decision. The superscript + indicates the maximum between a value and zero ($x^+ = \max(x, 0)$). Originally, ϵ was not present in LRP-z$^+$. We included it for numerical stability during Dual ISNet training (see later). Section 5 shows the beginning of the relevance propagation when explaining class c logit with LRP-z$^+$ [14].

B. ORIGINAL ISNet

Layer-Wise Relevance Propagation's goal is to interpret deep neural networks. The direct optimization of LRP heatmaps to improve a deep classifier's attention was recently proposed in the original ISNet paper [3]. The ISNet is defined by a training procedure named Background Relevance Minimization (BRM), which minimizes the background relevance magnitude in a classifier's LRP heatmaps. The ISNet Loss (L_{IS}) is a linear combination of two functions, a standard classification loss (e.g., cross-entropy) L_C and a heatmap loss L_{LRP}. Globally one has:

$$L_{IS} = (1 - P) \cdot L_C + P \cdot L_{LRP} \tag{6}$$

where the hyper-parameter $P \in (0, 1)$ rules the trade-off between background attention rejection and regular classification loss fitting. L_{LRP} (Equation (II-C)) utilizes training-time-known segmentation masks to identify the background in the classifier's LRP heatmaps, and it penalizes background relevance. The masks are images valued 1 over the foreground, and 0 in the background. They may be manually drawn or produced by pre-trained semantic segmenters, which can be application-specific (e.g., U-Net [16]), or general novel class segmentation DNNs (e.g., DeepMAC [17] or SAM [18]).

During the training phase, the main difference between an ISNet and a traditional DNN is that, in order to evaluate the ISNet's heatmap loss, one needs to create LRP

heatmaps of the current batch. In detail, one employs LRP-ε throughout the DNN, except for the first layer, which utilizes LRP-z^B. At inference time, the ISNet classifier can focus exclusively on the image foreground without the help of segmentation masks nor any auxiliary semantic segmentation DNN. The consequence is that the inference ISNet's architecture and computational cost are identical to a standard classifier's. However, the network's decision rules rely only on the image's foreground features, indicating it acquired an Implicit Segmentation skill, from which the name ISNet. Due to its resistance to background bias, the ISNet could hinder shortcut learning and improve generalization to out-of-distribution (o.o.d.) data [3].

1) ISNet: ADVANTAGES OF LRP OPTIMIZATION

In the original ISNet paper [3], we justified the choice of optimizing LRP both theoretically and empirically. Empirically, in all 5 applications with synthetic and non-synthetic background bias in [3], the ISNet background bias resistance and o.o.d. generalization significantly surpassed 7 state-of-the-art background bias mitigation DNN architectures: multi-task DNNs (performing classification and segmentation) [19], attention-gated neural networks (AG-Sononet) [20], Guided Attention Inference Networks (GAIN) [21], Hierarchical Attention Mining (HAM) [22], the standard segmentation-classification pipeline, vision transformers [23], and the Right for the Right Reasons (RRR) neural network [24]. Two of these architectures minimize backgrounds in explanation techniques other than LRP: RRR (input gradients) and GAIN (Grad-CAM). The ISNet also surpassed the ISNet Grad*Input, an ablation study where we substituted LRP by Gradient*Input [25] explanations in the ISNet. We presented the theoretical justification for the positive results of the ISNet LRP optimization in detail in [3], and we summarize it here:

- LRP satisfies the basic requirements for direct optimization by a loss function: differentiability and computational efficiency. The time required to create a LRP heatmap is similar to the time needed for a standard gradient backward pass [3].
- The precision of the ISNet's Implicit Segmentation is justified by the high level of abstraction and precise spatial information portrayed in LRP heatmaps [3]. The LRP propagation procedure considers all the DNN parameters and layers' activations, embedding in a single heatmap both the late layers' high-level semantics and context information, and the early layers' high-definition spatial information [3]. Thus, LRP avoids the trade-off between high resolution and high-level semantics, which exists in explanation techniques like Grad-CAM [26].
- LRP is principled: the LRP rules in this study and in the original ISNet (LRP-0, LRP-ε, LRP-z^B and LRP-z^+) are justified by the Deep Taylor Decomposition (DTD) framework [14]. Briefly, these rules propagate relevance according to a series of approximate Taylor expansions performed at the DNN neurons [14], capturing the influence that a neuron exerts on other neurons. The minimization of the LRP relevance flow to the images' backgrounds constrains the corresponding influence flow from background bias to the DNN's output. This justifies why the ISNet's decisions are robust to the effect of background bias [3].
- The ε hyper-parameter in LRP-ε is responsible for denoising the explanation heatmaps, thus improving the LRP loss convergence. First, higher ε reduces the first-order Taylor approximation error in LRP-ε, attenuating the LRP heatmaps' noise and improving their coherence and contextualization [14]. Second, [3] demonstrates that higher ε reduces the influence that weakly activated neurons have on the explanation heatmap, denoising it. Indeed, if $\varepsilon = 0$, LRP becomes equivalent to gradient*input heatmaps [3], which are noisy for large DNNs [27]. Due to the LRP-ε denoising properties, the ISNet Loss could stably and efficiently converge even for very deep backbones, unlike what was observed in Gradient*Input and input gradient optimization [3].

C. ISNet LOSS AND HYPER-PARAMETER SELECTION

The most important hyper-parameter for training the ISNet is P, which balances the influence of L_C and L_{LRP} in the loss. The DNN is sensible to this parameter as low values allow attention to background bias, while high P can reduce training speed. The original ISNet study presents strategies to tune P, considering access (or no access) to o.o.d. validation data [3]. The heatmap loss is composed of two terms, the heatmap background loss, L_1, and the heatmap foreground loss, L_2, as shown in eq. (7):

$$L_{LRP} = w_1 \cdot L_1 + w_2 \cdot L_2, \text{ where } 0 < w_1 \text{ and } 0 < w_2 \quad (7)$$

The first term is responsible for measuring background attention, while the second avoids a zero solution to L_{LRP} (when the neural network produces heatmaps valued zero everywhere), or exploding heatmap relevance. The two balancing hyper-parameters, w_1 and w_2, have little influence over the ISNet behavior. Thus, a fine search is not necessary to define them. We set both to 1 in the MNIST experiments, and we set $w_1 = 1$ and $w_2 = 3$ in the other applications, matching the values in the original ISNet [3]. However, in preliminary tests, we found similar results when setting both parameters to 1 in all applications. The calculation of the heatmap background loss involves a few processing steps, summarized below. For a detailed mathematical definition and justification for each procedure, please refer to the paper presenting the original ISNet [3].

1) Absolute heatmaps: take the absolute value of the LRP heatmaps.
2) Normalized absolute heatmaps: normalize the absolute heatmaps, dividing them by the absolute average relevance in their foreground.

3) Segmented heatmaps: in the normalized absolute heatmaps, set all foreground relevance to zero, by element-wise multiplying them by inverted segmentation targets (i.e., figures valued one in the background and 0 in the foreground).
4) Raw background attention scores: use Global Weighted Ranked Pooling [28] (GWRP) over the segmented heatmaps, obtaining one scalar score per map channel.
5) Activated scores: pass the raw scores through the non-linear function $f(x) = x/(x + E)$, where E is a constant hyper-parameter, normally set as 1.
6) Background attention loss: calculate the cross-entropy between the activated scores and zero, i.e., apply the function $g(x) = -ln(1 - x)$
7) L_1: calculate the average background attention loss for all heatmaps in the training mini-batch.

GWRP [28] (step 4 in the procedure) is a weighted arithmetic mean (in the two spatial dimensions) of the elements in the segmented heatmaps (output of step 3). GWRP ranks every element in the heatmaps in descending order. Following this order, the elements' weights in the summation decay exponentially, according to a hyper-parameter d, valued between 0 and 1. Thus, high relevance elements in the LRP heatmap background highly increment the heatmap loss. If $d = 0$, GWRP is equivalent to max pooling. When it is set as 1, GWRP is the same as average pooling. The original ISNet study suggested increasing d for improving training stability and reducing it to increase resistance to background bias (especially avoiding small high-attention regions in the background). The study also showed procedures to tune the d hyper-parameter, with or without access to o.o.d. validation data [3].

Finally, the second term of the heatmap loss, the foreground loss, L_2, is calculated with the procedure summarized below. Again, a detailed mathematical explanation is provided in the paper presenting the ISNet [3].

1) Absolute heatmaps: take absolute value of the LRP heatmaps.
2) Segmented maps: element-wise multiply the absolute heatmaps by the segmentation targets, which are valued 1 in the foreground and 0 in the background.
3) Absolute foreground relevance: sum all elements in the segmented maps, obtaining the total (absolute) foreground relevance per-heatmap.
4) Square losses: if the absolute foreground relevance, s, is smaller than a constant hyper-parameter, C_1, the correspondent square loss is $(C_1 - s)^2/C_1^2$. If it is larger than C_2 ($C_2 > C_1 > 0$), its correspondent square loss is $(C_2 - s)^2/C_1^2$. If $C_1 > s > C_2$, the correspondent loss is 0.
5) L_2: take the average of all square losses, considering all heatmaps in the mini-batch.

The L_2 loss is valued 0 when the heatmap absolute foreground relevance is in the range $[C_1, C_2]$, and it raises quadratically when it exits the range. The loss guarantees that the heatmap relevances will stay within a natural range, avoiding undesirable solutions to the background heatmap loss (L_1), such as solutions involving heatmaps valued zero or exploding heatmap elements.

D. ISNet LOSS MODIFICATION

Here, we propose a modification to the original ISNet loss, specifically, to L_2. We employed it in all ISNets trained in this study. The gradient of square losses in Step 4 of the procedure above can become exceedingly high if the absolute foreground relevance considerably exceeds the expected natural relevance range ($s \gg C_2$). This situation can happen in the beginning of the training procedure, especially if C_2 is small (e.g., due to weight decay, as explained in section II-E). Therefore, for training stability, we change the loss function in Step 4, switching from a square to a linear loss for high values of s. Equation (8) expresses the new loss function in Step 4, and fig. 1 plots it.

$$f(s) = \begin{cases} (C_1 - s)^2/C_1^2 & \text{if } s < C_1 \\ 0 & \text{if } C_1 \leq s \leq C_2 \\ (C_2 - s)^2/C_1^2 & \text{if } C_2 \leq s \leq C_2 + C_1 \\ 1 + s - (C_2 + C_1) & \text{if } C_2 + C_1 \leq s \end{cases} \tag{8}$$

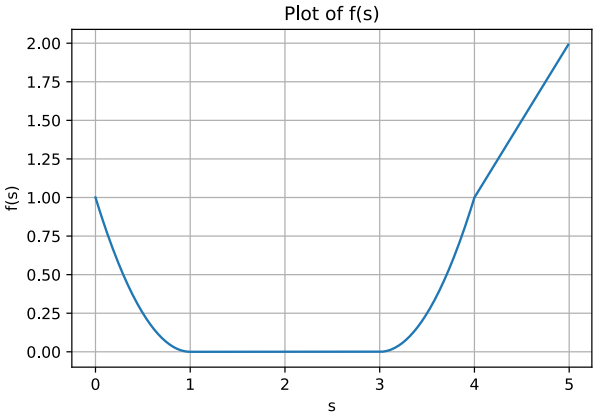

FIGURE 1. Plot of the f(s) function, for $C_1 = 1$ and $C_2 = 3$.

This loss function can be interpreted as the combination of the robust Huber loss and the ϵ-insensitive one for support vector machines together with a shift on the input.

E. HEATMAP RANGE SELECTION

The $[C_1, C_2]$ range objective is to approximate the natural interval of the total absolute relevance (sum of the absolute values of all heatmap elements). Therefore, it is derived from the analysis of heatmaps from a standard classifier (non-ISNet), which follows the ISNet backbone architecture, and shares its training dataset [3]. This natural range is similar for the same classifier architecture and heatmap creation procedure, even for diverse training datasets and classification tasks [3]. However, the ideal $[C_1, C_2]$ can drastically change when we alter the ISNet explanation strategy (e.g., changing from the original ISNet to the

Selective ISNet or ISNet Softmax Grad*Input), or the ISNet classifier backbone (e.g., from DenseNet to ResNet or VGG).

To accelerate hyper-parameter search, we have empirically derived a heuristic to define the hyper-parameters C_1 and C_2. First, train a standard classifier (non-ISNet) with the same architecture and training algorithm as the ISNet classifier, using the same training dataset, but for a small number of epochs (we used 4). Since weight decay can reduce the natural range of LRP relevance through training, we suggest training the non-ISNet network for longer when using the technique (up to the same number of epochs that will be used for the ISNet). Afterward, train the standard classifier for one epoch more, but generate its heatmaps for all training samples (using the same heatmap creation procedure that the ISNet will employ). For each heatmap, calculate the total absolute relevance ($\sum_j |H_j|$, where H_j is a heatmap element). During this last epoch, use Welford's online algorithm [29] to estimate the mean and standard deviation of the total absolute relevance across all training samples. The online method reduces memory consumption. Finally, define C_1 and C_2 according to the equations below, where M indicates the calculated mean and S the standard deviation:

$$C_1 = \max(M/5, M - 3S) \tag{9}$$

$$C_2 = \min(25C_1, M + 3S) \tag{10}$$

The hyper-parameters are defined according to absolute total relevance in a non-ISNet network, because we desire to understand what the natural range of values in a heatmap are. Since we are not interested in the standard classifier's ability to concentrate attention on the region of interest, we do not derive $[C_1, C_2]$ from the network's foreground absolute relevance. Instead, the ISNet optimization process will concentrate heatmap relevance inside the foreground, making the ISNet's foreground absolute relevance range approximate the standard classifier's total absolute relevance range.

Ideally, Equations 9 and 10 select the range $[M - 3S, M + 3S]$, which should capture a substantial portion of the total absolute relevance we observed for the standard classifier during training. However, we must use the maximum operation in eq. (9) to avoid C_1 becoming negative or too close to zero. Moreover, the minimum in eq. (10) avoids an exceedingly major difference between the two hyper-parameters, which could allow unnatural solutions for the heatmap loss [3]. The quantities $M/5$ and $25C_1$ were empirically defined during preliminary tests.

The natural range of relevance values can change across DNN layers, due to LRP relevance absorption [14]. Therefore, if we penalize LRP heatmaps at the inputs of multiple hidden layers (LRP Deep Supervision or LDS), we must define a C_1 C_2 pair for each of these layers. To do so, instead of only monitoring the standard classifier's input-level heatmaps, we calculate the mean (M) and standard deviation (S) of the total absolute LRP relevance independently (and in parallel) for all supervised layers. Accordingly, Equations 9 and 10 can provide the parameters $[C_1, C_2]$ for each penalized layer.

Instabilities in gradient backpropagation (e.g., vanishing and exploding gradients) can be harmful for LRP relevance propagation. If the model's input gradients are problematic around the data point (the input image), first-order Taylor expansions considering nearby reference points will probably also have issues. LRP is based on the approximation of a series of local Taylor expansions [14]. Thus, LRP heatmaps may also explode or vanish when gradients explode or vanish. In such cases, Equations 9 and 10 will return extremely high or low values, which are not adequate for training an ISNet. However, this problem can be solved by standard techniques that avoid vanishing and exploding gradients, such as adequate weight initialization and batch normalization.

F. FASTER ISNet

When one starts the LRP propagation from a given class score (logit), the resulting heatmap explains how each input element influenced that logit. Therefore, the map may not properly capture the effect of background bias on logits representing other classes. The standard LRP procedure is to explain the highest logit, which represents the winning class [14]. In this case, the resulting heatmap may not show attention to a background bias that reduced the losing classes' logits. However, this background bias can alter the classifier's decision. Accordingly, minimizing background relevance only in the winning classes heatmaps could not hinder background bias attention [3].

To account for the background influence over all class scores, the original ISNet generates and optimizes, for each training image, one heatmap explaining each possible logit. The multiple heatmaps can be processed in parallel (like batch samples). However, considering a classification task with C categories, ISNet's training time increases approximately linearly with C. Memory consumption also increases, and, when it reaches the available limit, the heatmaps need to be produced serially, increasing training time linearly with C [3].

Here, we present three ISNet reformulations, which use alternative LRP procedures to enable the creation of only one, or two, heatmaps per training image. Therefore, the here proposed architectures remove the dependency of the training time (or memory consumption) on C. Accordingly, we collectively call them ''Faster ISNet''. We introduce three variants: Dual ISNet, Selective ISNet and Stochastic ISNet.

1) DUAL ISNet

To avoid creating and penalizing multiple heatmaps, we should ideally produce and optimize a single LRP map that reveals how input features influence all the classifier's class scores (logits) jointly. We can propagate relevance from all logits simultaneously instead of propagating it from a single logit z_c^{Lmax}. Accordingly, instead of using eq. (3) to define the logit's relevances, we set each logit

relevance as the logit value ($R_k^{\text{Lmax}+1} = z_k^{\text{Lmax}} \forall k \in [1, 2, \ldots, C]$). Then, we propagate these relevances with the standard LRP-ε rule (and LRP-z^B in the first DNN layer, optionally, Appendix E). We dubbed the resulting explanation the joint LRP-ε heatmap. Creating a joint heatmap takes the same time as creating a standard map. Unfortunately, during the relevance propagation, positive background relevance originating from a logit may encounter negative background relevance that originated from another logit, causing destructive interference in the joint LRP-ε heatmap. This phenomenon can deceivingly reduce the amount of background relevance in the joint heatmap. Hence, it is not reliable to solely base the ISNet Loss on joint LRP-ε heatmaps.

As the LRP-z^+ rule only propagates positive relevance, the joint LRP-z^+ map is not subject to destructive interference. To produce this heatmap, we set the logits' relevances as $R_k^{\text{Lmax}+1} = \sum_j (w_{jk}^{\text{Lmax}} a_j^{\text{Lmax}})^+ \forall k \in [1, 2, \ldots, C]$, instead of using eq. (5). Then, we further propagate the relevance with the standard LRP-z^+ rule (eq. (4)), except for the first DNN layer, where may use LRP-z^B. Optimizing solely the joint LRP-z^+ map can also be insufficient to minimize the influence of background bias on the classifier. By not propagating negative relevance, the LRP-z^+ explanation ignores negative evidence (section (II-A)), which may also cause shortcut learning [3].

To ensure background bias resistance, the Dual ISNet creates two LRP heatmaps per training image: the LRP-ε joint map, and the LRP-z^+ joint map. Both are individually penalized by the heatmap loss. As LRP-z^+ is immune to destructive interference, the background relevance minimization on the LRP-z^+ joint map will not allow the image background to positively contribute to the DNN logits. Consequently, LRP-z^+ optimization also minimizes positive background relevance in the joint LRP-ε heatmap, avoiding destructive interference in the map's background. Accordingly, background relevance minimization in the LRP-ε map becomes able to hinder negative evidence in the image's background. Therefore, the Dual ISNet minimizes the influence of the background on the classifier decisions. Moreover, it introduces a potentially massive training speed improvement, by replacing the C (number of classes) LRP-ε heatmaps in the original ISNet by only two joint heatmaps. Appendix B explains a fast procedure to compute LRP-z^+ and LRP-ε joint heatmaps, which we included in the LRP Block (the ISNet LRP implementation).

2) SELECTIVE ISNet

The Selective ISNet produces and optimizes one LRP heatmap per training sample. To ensure background bias resistance, the map must capture the influence of the image background over all logits. Thus, the Selective ISNet is defined by the following reformulation of the ISNet LRP procedure: instead of creating C LRP heatmaps to explain the C DNN logits (z_c^{Lmax}, where $c \in [1, C]$), create a single LRP heatmap that explains a Softmax-based quantity η_c (eq. (11)), where c is the class associated to the lowest DNN logit.

$$\eta_c = \ln\left(\frac{P_c}{1 - P_c}\right) \tag{11}$$

where:

$$P_c = \frac{e^{z_c^{\text{Lmax}}}}{\sum_{c'=1}^{C} e^{z_{c'}^{\text{Lmax}}}} \tag{12}$$

η_c is a monotonically increasing function of P_c, the Softmax predicted probability of class c (eq. (12)). P_c depends on all DNN's logits. Thus, the heatmap explaining η_c depends on how the input features affect all class scores; e.g., a background bias that only reduces the losing classes' logits will consequently increase the classifier confidence for the winning class (P_h). Thus, the bias will increase η_h. Accordingly, it produce positive background relevance in the LRP heatmap explaining η_h, even though the bias does not appear in the heatmap explaining the logit z_h^{Lmax}. The explanation of η_c instead of logits was originally proposed to make LRP heatmaps more class-selective [14]. Indeed, η_c heatmaps better reveal how input features affected the classifier confidence for class c (P_c). For example, in a heatmap explaining a logit z_c^{Lmax}, an input feature can be represented as positive relevance for having increased z_c^{Lmax}. However, the feature may have actually reduced P_c, by strongly incrementing the other logits. In this case, the feature will reduce η_c, and be represented as negative relevance in the η_c heatmap.

To explain η_c instead of the logit z_c^{Lmax}, we only change the LRP propagation rule for the last DNN layer. Instead of using the procedure in section II-A, we use eq. (13) to obtain the LRP relevance at the input of the last layer (R_j) [14]:

$$R_j^{\text{Lmax}} = \sum_{c'=1}^{C} \frac{(w_{jc}^{\text{Lmax}} - w_{jc'}^{\text{Lmax}}) a_j^{\text{Lmax}}}{z_{c,c'} + \text{sign}(z_{c,c'})\epsilon} R_{c,c'} \tag{13}$$

where:

$$R_{c,c'} = \frac{z_{c,c'} e^{-z_{c,c'}}}{\sum_{c'' \neq c} e^{-z_{c,c''}} + \mu} \tag{14}$$

$$z_{c,c'} = z_c^{\text{Lmax}} - z_{c'}^{\text{Lmax}} = \sum_j a_j^{\text{Lmax}} (w_{jc}^{\text{Lmax}} - w_{jc'}^{\text{Lmax}}) \tag{15}$$

Then, the relevance is further propagated with LRP-ε (and LRP-z^B in the input layer, optionally, Appendix E). In the equation, a_j^{Lmax} represents the DNN last layer inputs, w_{jc}^{Lmax} is the weight connecting input a_j^{Lmax} to logit z_c^{Lmax} (w_{0c}^{Lmax} is a bias parameter, and $a_{0c}^{\text{Lmax}} = 1$), ϵ is the LRP-ϵ stabilizer hyper-parameter (set to 10^{-2}), and μ is a small positive constant (10^{-5}), which we add to improve numerical stability. The equation is defined for a fully-connected last layer, but it is also applicable to other layers that can be expressed as a fully-connected layer (e.g., convolution). Appendix B shows how we altered the ISNet LRP implementation (the LRP

Block) to apply eq. (13) in an computationally convenient manner.

η_c is an unbounded quantity, whose magnitude increases with the difference between P_c and 0.5, where $\eta_c = 0$. Thus, when we start the LRP procedure from η_c, small and high P_c normally produce heatmaps containing higher relevance magnitude. Indeed, $P_c = 0.5$ represents a state of maximal classifier uncertainty for class c (50% probability), justifying why LRP relevances should be closer to zero. During the beginning of the training procedure, it is common for the highest class probability (P_c) to be near 50%. However, it is rarer for the lowest P_c to be near 0.5, especially with multiple possible classes. Therefore, to create more expressive heatmaps, we chose to explain η_c for the class c corresponding to the lowest DNN logit (for each training image).

3) STOCHASTIC ISNet

As previously discussed, the heatmap explaining a single class logit cannot properly capture how bias affects all class scores. Thus, the standard practice of producing LRP heatmaps for the winning class or the label class [14] is not adequate for background relevance minimization. However, we hypothesize that the minimization of background relevance in random class (c) heatmaps may avoid background bias attention.

A trivial strategy would be choosing c from an uniform probability distribution for each training image, then creating and optimizing the LRP-ε map that explains logit c. However, assigning the same probabilities for all classes could be problematic. Only the map explaining the highest logit can properly show positive correlations between the winning logit and the presence of background bias. Such correlations can strongly affect the classifier decisions and must be effectively penalized. Let us consider an unbalanced classification dataset with C categories, where class A has a small number of samples, N. Moreover, assume that the classifier being trained already has high accuracy. In this case, following the random uniform selection of logits, only about N/C winning logit heatmaps (explaining class A) will be created for the class A images in one epoch. If N is small and C is large, a positive correlation between background bias and the class A logit will have a small effect in the average heatmap loss. Thus, background bias attention will not be effectively hindered for class A. For this reason, we devised the following logit selection strategy: give the highest logit 50% selection probability, while all other logits have (50/(C-1))% probability.

Using this strategy, the Stochastic ISNet selects one logit per training image and explains it with a LRP-ε heatmap (using LRP-z^B in the first DNN layer, optionally, Appendix E), which the heatmap loss optimizes. During training, the technique creates approximately the same number of winning and losing class heatmaps. As previously explained, the optimization of winning class heatmaps are important to hinder positive correlations between background bias and the winning logits. Meanwhile, the penalization of losing class heatmaps prevents the background bias from acting as negative evidence and influencing the classifier decision by reducing the losing class logits. The Stochastic ISNet requires no modifications to the original ISNet LRP implementation (LRP Block).

G. LRP-FLEX: A SIMPLE, FAST AND MODEL-AGNOSTIC IMPLEMENTATION OF LRP

Layer-Wise Relevance Propagation can explain virtually any DNN architecture [14]. However, specific coding is normally required for each different architecture. Moreover, LRP libraries are commonly large and complex, especially when implementing LRP for multiple architectures. The original ISNet LRP Block [3] implemented the rules LRP-ε and LRP-z^B for the DenseNet [10], VGG [30] and simple sequential networks (defined as a PyTorch Sequential object), using about 2000 lines of code. With our inclusion of the LRP-z^+ rule and the LRP ResNet implementation, it now has about 4000 lines.

We introduce LRP-Flex, an implementation of LRP-ε for DNNs utilizing only ReLU nonlinearities in their hidden layers (the last layer can have alternative activations). Its key features are: first, it is exceedingly simple, requiring significantly less code lines than standard LRP implementations (e.g., the LRP-Flex PyTorch code is about 10 times shorter than the LRP Block); second, it is model-agnostic, being readily applicable to arbitrary DNN architectures, and not requiring the user to spend time writing architecture-specific code; third, it is fast, taking advantage of highly optimized backpropagation engines available in deep learning libraries. LRP-Flex produces differentiable heatmaps. Thus, it can be employed to easily implement the original ISNet, the Stochastic ISNet, and the Selective ISNet for any ReLU-based classifier architecture. Furthermore, it is a practical and fast technique to explain DNN's decisions with LRP. We summarize LRP-Flex workflow below. The algorithm is based on an equivalent reformulation of LRP, which is elucidated in Appendix A. The workflow considers the LRP-ε rule, but it can be easily expanded to use other rules in specific DNN layers (Appendix A); e.g., we may use LRP-z^B for the first DNN layer.

1) Initialization: modify the gradient backpropagation procedure for all ReLU functions in the neural network (e.g., using PyTorch's backward hooks). The equation below defines the modified ReLU backpropagation rule. $\mathbf{G^{out}}$ (with elements G_k^{out}) is the quantity that was back-propagated until the output of the ReLU function ($\mathbf{a^{out}}$, with elements a_k^{out}). The equation back-propagates $\mathbf{G^{out}}$ to the input of the ReLU, producing $\mathbf{G^{in}}$ (with elements G_j^{in}). The variable ε is the small positive hyper-parameter in LRP-ε (e.g., 0.01). This backward pass modification has to be deactivated when not creating LRP heatmaps (e.g., for loss gradient

calculation).

$$G_k^{in} = \frac{a_k^{out}}{a_k^{out} + \varepsilon} G_k^{out}$$

2) Forward pass: run the neural network and store the outputs of its ReLU functions (e.g., employing forward hooks in PyTorch).
3) Modified backward pass: to explain the DNN logit z_c^{Lmax} (or η_c, for the Selective ISNet), request the automatic backpropagation engine in the deep learning library to calculate the gradient of z_c^{Lmax} (or $\tilde{\eta_c}$) with respect to the network's input ($\mathbf{X}$). However, use the modified backward procedure in all ReLU functions (Step 1). Accordingly, the resulting tensor ($\mathbf{G^0}$) will not match the actual input gradient, $\mathbf{G^0} \neq \nabla_{\mathbf{X}} z_c^{\text{Lmax}}$ (or $\mathbf{G^0} \neq \nabla_{\mathbf{X}} \tilde{\eta_c}$). $\tilde{\eta_c}$, defined below, is a bounded version of η_c (eq. (11)). μ is a small positive hyper-parameter (e.g., 0.01), which ensures numerical stability, and P_c is the softmax-estimated class c probability (eq. (12)).

$$\tilde{\eta_c} = \ln(P_c + \mu) - \ln(1 - P_c + \mu)$$

4) Element-wise multiplication: to obtain the final LRP-ε heatmap ($\mathbf{R^0}$), element-wise multiply the back-propagated quantity ($\mathbf{G^0}$) and the DNN input ($\mathbf{X}$).

$$\mathbf{R^0} = \mathbf{G^0} \odot \mathbf{X}$$

Algorithm 1 LRP Deep Supervision (LDS) - Forward Pass

1: Classify training batch
2: Perform LRP once, save the relevance signal at the input of each layer selected for deep supervision, H_L
3: **for** each selected layer L **do**
4: Resize foreground segmentation mask to fit H_L
5: Compute heatmap loss $L_{LRP,L}$ for H_L and mask
6: **end for**
7: Use GWRP to aggregate multiple losses $L_{LRP,L}$ into a scalar loss, L_{LRP}
8: Calculate the ISNet loss $L_{IS} = (1 - P) \cdot L_C + P \cdot L_{LRP}$

H. LRP DEEP SUPERVISION

During relevance propagation, LRP produces intermediate heatmaps at each DNN depth. They explain how hidden layers' inputs influenced the DNN output. Since all these heatmaps portray a high level of abstraction (which LRP carries from deep layers), we are able to minimize their background relevance. Here, we introduce LRP Deep Supervision or LDS (algorithm 1), a technique that leverages intermediate LRP heatmaps to improve the ISNet convergence. It resizes foreground segmentation masks to fit multiple intermediate heatmaps and applies the ISNet's heatmap loss to them. All losses use the same hyper-parameters, except for C_1 and C_2, which we automatically set per-layer (section II-E). To aggregate the multiple losses into a scalar loss, we use Global Weighted Ranked Pooling (GWRP): we arrange the losses in descending order, give them exponentially decreasing weights, and perform a weighted average [28]. Thus, GWRP prioritizes high losses, ensuring none of the supervised heatmaps has significant background attention. Intermediate heatmap losses are auxiliary objectives to improve the convergence of the standard (input-level) heatmap loss.

Globally, LDS constrains the DNN to discard information from both the input's and the feature maps' backgrounds. Thus, it enforces direct mapping, encouraging feature maps' foregrounds to portray only relevant information, extracted solely from the DNN input's foreground. The L^{th} layer's LRP heatmap expresses the attention of a sub-network, defined by layer L and all subsequent layers. Locally, LDS guides multiple sub-networks to focus on the foregrounds of their input feature maps. Optimizing the attention of a smaller sub-network should be easier than optimizing the entire DNN's focus. Moreover, the improvement of a sub-network's attention profile should help the optimization of larger sub-networks. Hence, we expect LDS to improve ISNet loss convergence in difficult optimization problems, increasing robustness and accuracy. To reduce computational cost, LDS supervises only a few DNN layers. We prioritize layers representing signal bottlenecks, i.e., layers that are not in parallel with skip connections (Appendix E).

III. RESULTS

First, we perform experiments on MNIST and natural images (Stanford Dogs) with synthetic background bias, to quantitatively assess the Faster ISNet robustness to shortcut learning. Afterward, we experiment on a mixed source chest X-ray dataset, presenting non-synthetic background bias. This task represents a contemporary application where background bias is common [2]. Datasets are detailed in Appendix G, data processing in D, benchmark architectures in C, and training procedure in E.

TABLE 1. Test results for the neural networks trained on MNIST with synthetic background bias. The i.i.d. test data contains the same bias as the training dataset, the o.o.d. test database has no background bias, and the deceiving bias test has bias designed to fool the classifiers. The reference classifier is the only model trained on a dataset without synthetic bias. Our models (Faster ISNet) are highlighted in bold text. On this low-resolution dataset, with classifiers based on a ResNet18, RRR and all ISNets were accurate and robust to bias.

Neural Network	ACC i.i.d.	ACC o.o.d.	ACC Deceiving
Standard Classifier	1	0.654	0.124
Dual ISNet	0.979	0.979	0.979
Selective ISNet	0.967	0.967	0.967
Stochastic ISNet	0.983	0.983	0.983
ISNet Softmax Grad*Input	0.949	0.949	0.949
Original ISNet	0.985	0.985	0.985
RRR	0.97	0.967	0.967
Reference Classifier	-	0.985	-

A. SYNTHETICALLY BIASED MNIST

Table 1 reports results on the synthetically biased MNIST [12], where training images contain white background pixels whose positions correlate to the image classes (digits). In this

application, DNNs use a ResNet18 backbone. We also implemented a reference classifier, a ResNet18 trained without synthetic bias. In synthetic bias applications, we considered 3 versions of the test set: i.i.d., containing the synthetic bias akin to the training set; o.o.d., with no bias; and deceiving bias, where we changed the correlations between biases and image classes to fool the classifiers (e.g., we take a bias that was correlated to MNIST digit 3 during training and associate it to digit 4 in the deceiving bias test set). The test accuracy reduction when synthetic bias is removed or replaced by confounding bias is a quantitative measurement of background bias influence on DNNs, quantifying shortcut learning. The standard classifier (ResNet18) reveals the extreme tendency for shortcut learning in the biased MNIST dataset: its accuracy drops from 100% to 65.4% and to 12.4% when the bias is removed or substituted by confounding bias, respectively. RRR [24] was robust, with 0.3% accuracy drop upon bias removal. All ISNet variants had similar results and were resistant to background bias, displaying no accuracy drop across the 3 test settings. Also, they were accurate, performing similarly to a DNN trained without bias. Multiple neural networks used as benchmark in the original ISNet paper [3], such as GAIN and the multi-task U-Net, are not adequate for the MNIST dataset: its small resolution images can be incompatible with large architectures, and its feature maps may be too minute for explanations like Grad-CAM. Conversely, with LRP-Flex, the Faster ISNet can readily accept any ReLU-based architecture, and LRP produces adequate explanations even when late feature maps are too tiny for Grad-CAM.

Figure 2 presents LRP heatmaps for biased MNIST test images (i.i.d. test). Thus, the images contain the same synthetic background bias found in training data (the white pixels in the images' upper region). Figure 2 heatmaps demonstrate that the bias could not influence the Faster and Original ISNets. The ISNet Softmax Grad*Input shows minimal bias attention, only in the image displaying a 5. RRR paid some attention to bias in 2 of the 4 images, while the standard classifier mostly focused on bias. Such findings corroborate with the quantitative results in Table 1, which prove that the synthetic bias had no influence over the Faster and Original ISNet decisions, it had a small influence over RRR, and major influence over the standard ResNet18. Although Table 1 shows no accuracy drop upon bias removal for the ISNet Softmax Grad*Input, there is a very small drop when we analyze results with one additional significant figure: its accuracy was 0.949 in i.i.d. testing, 0.9486 in o.o.d., and 0.9485 in confounding bias. The 0.04-0.05% accuracy drop explains the tiny attention to bias the network demonstrates in fig. 2.

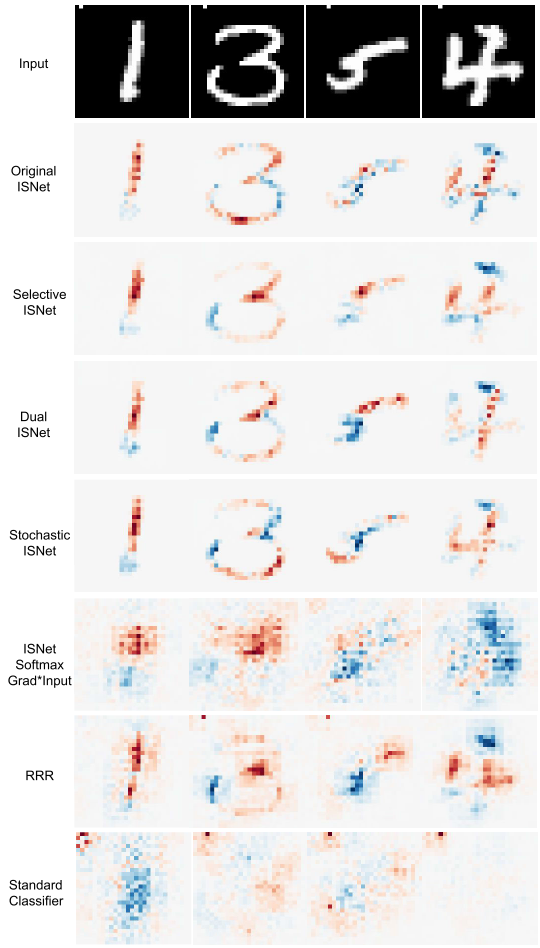

FIGURE 2. LRP heatmaps for the MNIST classification task. A white pixel in the images' top left corner represents synthetic background bias. Red colors indicate areas that contributed for the DNN confidence in the images' true class, while blue areas reduced such confidence. White regions were not important for the DNN decision. All LRP heatmaps were created with LRP-ε throughout the DNN, and LRP-z^B for the input layer. Red colors (positive relevance) indicate areas that the DNN associated with the ground-truth class. Negative relevance is shown in blue, representing regions that decreased the classifier's confidence in the ground-truth class. White regions were not important for the classifier's decisions (low absolute relevance). The ISNet and Faster ISNet loss objective is the minimization of background attention, while foreground attention is guided by the standard classification loss. Accordingly, the ISNets paid no attention to background bias.

B. SYNTHETICALLY BIASED STANFORD DOGS

Table 2 reports results (macro averaged one-vs-one ROC-AUC [31]) for the synthetically biased Stanford Dogs dataset. This dataset presents a difficult fine-grained classification problem, with 120 dog breeds (classes), small inter-class variation, and large intra-class variation [13]. Here, the synthetic bias is a number representing the dog breed, added to the image's background. All DNNs use a ResNet50 backbone, except for vision transformer [23], Multi-task U-Net [3] and AG-Sononet [20] (Appendix C). As Stanford Dogs has 120 classes, we did not train the original ISNet on it; it would take about 2 months, while Faster ISNets required about 1 day (section III-E). In Stanford Dogs, the synthetic bias also caused a strong shortcut learning tendency, demonstrated by the standard classifier's (ResNet50) strong performance drop upon bias removal or substitution by deceiving bias. Only the ISNets, GAIN (Grad-CAM optimization) [21], and

TABLE 2. Test results for the neural networks trained on Stanford Dogs with synthetic background bias. The i.i.d. test data contains the same bias as the training dataset, the o.o.d. test database has no background bias, and the deceiving bias test has bias designed to fool the classifiers. The reference classifier is the only model trained on a dataset without synthetic bias. Our models (Faster ISNet) are highlighted in bold text. Due to the number of classes in the dataset, the original ISNet could not be feasibly trained for this task, as training would take months. Faster ISNets could achieve high accuracy and bias robustness, especially the Dual ISNet and the ISNets with LDS. These models could even match a neural network classifying images whose backgrounds were removed (U-Net+Classifier).

Neural Network	AUC i.i.d.	AUC o.o.d.	AUC Deceiving
Standard Classifier	1	0.547	0.454
Dual ISNet	0.905	0.905	0.905
Selective ISNet	0.799	0.798	0.799
Stochastic ISNet	0.859	0.857	0.857
Dual ISNet LDS	0.895	0.895	0.895
Selective ISNet LDS	0.9	0.9	0.9
Stochastic ISNet LDS	0.885	0.885	0.885
ISNet Softmax Grad*Input	0.55	0.55	0.55
RRR	0.941	0.569	0.465
U-Net+Classifier	0.896	0.897	0.897
Multi-task U-Net	0.972	0.824	0.631
AG-Sononet	1	0.507	0.419
Extended GAIN	0.804	0.803	0.803
Vision Transformer	1	0.578	0.46
Reference Classifier	-	0.809	-

the segmentation-classification pipeline (U-Net+ResNet50) were robust to background bias and shortcut learning, being the only architectures whose performances remained effectively the same across the 3 test scenarios. However, GAIN was overshadowed by the pipeline's results. Conversely, some Faster ISNets could even surpass the pipeline. Thus, they did not trade accuracy for robustness. Instead of penalizing LRP, RRR and the ISNet Softmax Grad*Input ablation (Appendix C-A) minimize backgrounds in input gradients and Gradient*Input [25] heatmaps, respectively. With large images (e.g., 224 × 224) and deeper backbones (e.g., ResNet50), these 2 explanations become noisy [3]. Thus, the ISNet Softmax Grad*Input did not effectively converge in Stanford Dogs. Meanwhile, RRR minimized its less restrictive loss function [24] by reducing its input gradients' magnitude, instead of promoting foreground (dogs) focus. The same phenomena were observed and justified in [3].

We chose biased Stanford Dogs to evaluate LDS due to the higher difficulty of its optimization task, considering strong background bias and 120 classes. Showing that LRP optimization is indeed more difficult in this dataset, the Selective and Stochastic ISNets performances lagged behind the segmentation-classification pipeline. However, LDS, created to improve LRP loss convergence, allowed the models to match the pipeline. LDS had little effect on the Dual ISNet. This network already matched the pipeline without LDS, indicating that it was already effectively converging, making LDS unnecessary for the model.

Figure 3 presents the heatmaps for the synthetically biased Stanford Dogs i.i.d. test dataset. It demonstrates that, except for the Faster ISNet variants (fig. 3 rows 2 to 7), the segmentation-classification pipeline (row 9), and GAIN (row 11), all other neural networks paid significant attention

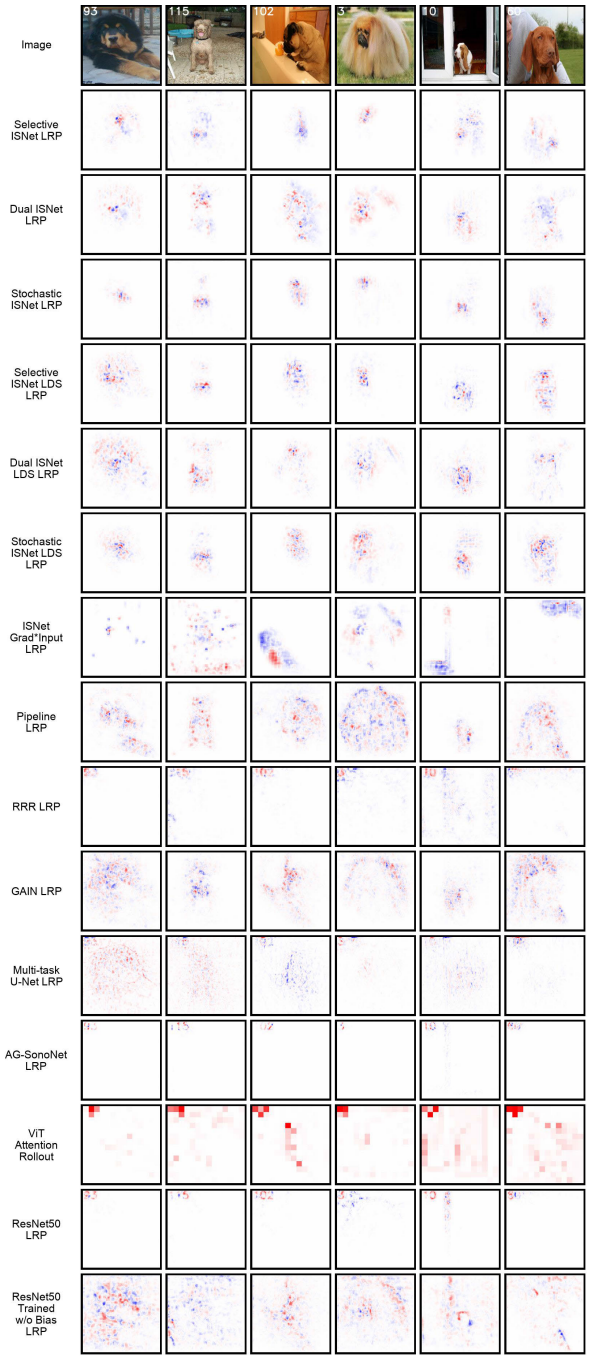

FIGURE 3. Heatmaps for the Stanford Dogs classification task. Numbers in the images' top left corner represent synthetic background bias. Red colors indicate areas that contributed for the DNN confidence in the images' true class, while blue areas reduced such confidence. White regions were not important for the DNN decision. Numbers in the images' top left corner indicate the dogs' breeds, representing synthetic background bias. All neural networks are explained with LRP, except for the Vision Transformer, which is explained with attention rollout, the standard explanation methodology for the architecture [23]. For reference, the last row in the figure presents attention maps for the ResNet50 trained and tested without synthetic background bias. Only the Faster ISNets, segmentation-classification pipeline and GAIN paid no attention to background bias.

to the numbers in the figures' backgrounds. Therefore, the heatmaps in the figure corroborate with the numerical results in Table 2, which quantitatively prove that the three aforementioned models were the only ones whose decisions were not influenced by background bias.

TABLE 3. Test AUC and F1-Scores (per class and average) for the external (out-of-distribution) COVID-19 X-ray database. Results on X-rays from hospitals never seen during training. F1-Scores are presented as mean±std, [HDI]. Mean refers to the metric's mean value, considering its probability distribution, and std refers to its standard deviation. The 95% high density interval (HDI) is an interval containing 95% of the metric's probability mass. Moreover, any point inside the interval has a higher probability density than any point outside of the interval. Please refer to Appendix F for more details about our statistical analysis. Our models (Faster ISNet) are highlighted in bold text. The Faster ISNet matched the original ISNet as the algorithms with the highest out-of-distribution performance.

Neural Network	Normal F1-Score	Pneumonia F1-Score	COVID-19 F1-Score	Macro-average F1-Score	Macro OvO AUC
Selective ISNet	0.625±0.017, [0.59,0.658]	0.783±0.01, [0.764,0.802]	0.902±0.006, [0.891,0.913]	0.77±0.008, [0.754,0.786]	0.946
Dual ISNet	0.534±0.018, [0.498,0.57]	0.731±0.01, [0.71,0.751]	0.881±0.006, [0.868,0.893]	0.715±0.009, [0.699,0.733]	0.909
Stochastic ISNet	0.604±0.018, [0.568,0.639]	0.709±0.011, [0.687,0.731]	0.879±0.006, [0.867,0.891]	0.731±0.009, [0.714,0.748]	0.935
ISNet Softmax	0.273±0.015, [0.244,0.302]	0.09±0.01, [0.07,0.111]	0.604±0.01, [0.585,0.623]	0.323±0.007, [0.308,0.337]	0.608
Grad*Input Original ISNet	0.555±0.022, [0.512,0.597]	0.858±0.007, [0.844,0.871]	0.907±0.006, [0.896,0.918]	0.773±0.009, [0.755,0.791]	0.952
Standard Classifier	0.444±0.02, [0.403,0.482]	0.434±0.015, [0.405,0.463]	0.76±0.008, [0.744,0.775]	0.546±0.01, [0.527,0.565]	0.808
U-Net+ Classifier	0.571±0.018, [0.535,0.607]	0.586±0.013, [0.561,0.611]	0.776±0.008, [0.76,0.792]	0.645±0.009, [0.626,0.663]	0.833
Multi-task U-Net	0.419±0.025, [0.369,0.469]	0.119±0.011, [0.098,0.14]	0.585±0.009, [0.566,0.602]	0.374±0.01, [0.355,0.394]	0.553
AG-Sononet	0.124±0.015, [0.096,0.153]	0.284±0.015, [0.255,0.312]	0.659±0.009, [0.641,0.676]	0.356±0.008, [0.34,0.372]	0.591
Extended GAIN	0.203±0.019, [0.166,0.24]	0.485±0.013, [0.46,0.511]	0.711±0.009, [0.693,0.728]	0.466±0.009, [0.449,0.485]	0.724
RRR	0.36±0.018, [0.325,0.394]	0.552±0.013, [0.526,0.577]	0.737±0.009, [0.72,0.755]	0.55±0.009, [0.532,0.568]	0.775
Vision Transformer	0.382±0.017, [0.348,0.415]	0.474±0.013, [0.448,0.499]	0.525±0.011, [0.503,0.548]	0.46±0.009, [0.443,0.478]	0.683

C. COVID-19 DETECTION IN CHEST X-RAYS

Without synthetic bias, we trained DNNs to classify COVID-19, pneumonia and normal on the mixed X-ray dataset from [3]. It has an o.o.d. test partition (X-rays from external hospitals), and improvements in o.o.d. performance reflect increased background bias robustness and shortcut learning reduction [1], [3]. Here, foreground means lungs. The task's results in table 3 are in line with past COVID-19 detection studies employing o.o.d. validation [2], [3], [32]. Conversely, i.i.d. COVID-19 detection performances can be deceitfully high and biased [2], [7]. We compared the Faster ISNet to the original ISNet and all benchmark DNNs trained in [3] for COVID-19 detection. The DenseNet121 was the backbone for ISNets and most benchmark DNNs [3], except for vision transformer [23], Multi-task U-Net [3] and AG-Sononet [20] (Appendix C). The Faster ISNet matched the original ISNet, and significantly surpassed all remaining state-of-the-art DNNs in the table.

Figure 4 presents LRP heatmap examples for the o.o.d. COVID-19 detection dataset. Please refer to the original ISNet study [3] for heatmaps representing the remaining neural networks (RRR, GAIN, multi-task DNN, AG-Sononet, vision transformer, and segmentation-classification pipeline). All ISNet variants focused exclusively on the lungs, as intended. There are lung regions that appear red for one network, and blue for another one. This finding indicates that neural networks may have difficulty differentiating damage caused by pneumonia and COVID-19, considering the similarity between the diseases [3]. Another explanation for the change in relevance sign may be insufficient class selectivity in LRP [33], which may have not perfectly captured if lesions were mostly associated to COVID-19 or pneumonia. However, perfect class selectivity is not important for the ISNet: the model minimizes the LRP relevance magnitude in the image background, independent of the relevance sign. Moreover, the foreground attention patterns in Figure 4 depend on the optimization of the classification loss, which is influenced by parameter initialization. Conversely, the ISNet heatmap loss goal and scope are avoiding background bias attention, not controlling how the DNN analyzes the foreground.

X-ray heatmaps revealed considerable background bias (e.g., text and marks) attention for all DNNs, except for the original ISNet, Faster ISNets, and segmentation-classification pipeline. A reduction of background bias' influence over the classifier indicates a reduction of shortcut learning, resulting in superior o.o.d. test performance [1]. Accordingly, the LRP-based ISNets significantly surpassed the remaining neural networks on the external (o.o.d.) COVID-19 test dataset. Hence, the quantitative results in table 3 support the information in the heatmaps presented in fig. 4 and in the original ISNet study [3]. Overall, LRP heatmaps and numerical results (i.e., o.o.d. performance and accuracy drop upon synthetic bias removal) show that the LRP-based ISNets and the segmentation-classification pipeline are the only implemented models that were consistently resistant to background bias in all experiments in this study and in [3]. Although GAIN was robust to bias in Stanford Dogs, it suffered spurious mapping in COVID-19 classification. Spurious mapping [3] happens when a classifier learns to produce deceiving Grad-CAM heatmaps, by mapping features from the background of an image to the foreground of the late DNN feature maps. This undesirable solution allows a DNN to minimize GAIN's losses, while still having decision rules that consider background bias. In the paper introducing the ISNet, this problem appeared when GAIN was trained for multiple tasks that considered background bias, including the COVID-19 detection problem that we also consider in this work [3]. Grad-CAM optimization may unpredictably lead to spurious mapping, but LRP optimization is immune to it [3]. All Faster ISNets performed comparably to the original ISNet, significantly surpassing the segmentation-classification pipeline, whose decisions considered shortcuts like erased backgrounds and lung borders [3].

D. COMPARISONS TO BENCHMARK MODELS

The original ISNet study justified why the ISNet surpassed the benchmark DNNs [3]. We summarize these findings here,

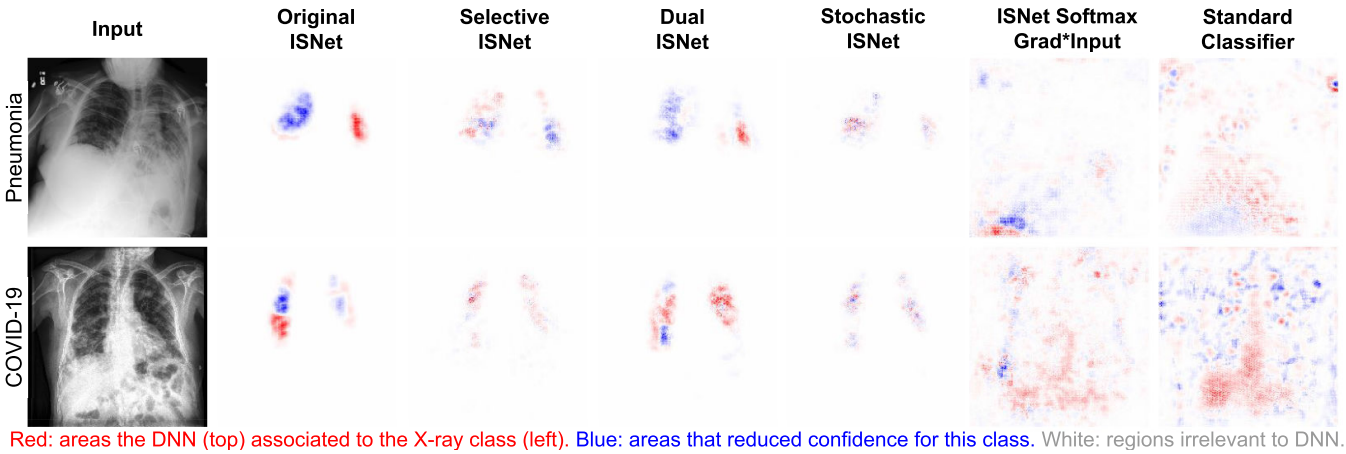

FIGURE 4. LRP heatmaps for X-rays. The pneumonia X-ray has a clear background bias, a mark over the right shoulder. It did not influence the Faster or Original ISNets. Heatmaps for the remaining benchmark architectures are available in [3], and background bias (e.g. text and markings) attention is noticeable for all DNNs except for the original ISNet, Faster ISNets and the large segmentation-classification pipeline [3].

and show that the results in this study confirm the analysis in [3]. We advise the interested reader to check [3] for a more detailed analysis and justification of each finding.

- **Segmentation-classification pipeline:** In [3], we found that the classifier in the pipeline could pay attention to the erased background and to the foreground's borders, indicating biased decision rules, which consider the foreground shape and position. Such rules can impair o.o.d. generalization, possibly explaining why the ISNet surpassed the pipeline in [3]. **Here**, the Faster ISNets and the original ISNet significantly surpassed the segmentation-classification pipeline in COVID-19 detection, a task where the model displayed attention to foreground (lung) borders and erased backgrounds [3]. This background attention was less noticeable in Stanford Dogs (fig. 3), where the pipeline could match the ISNets.
- **Multi-task U-Net:** In [3], we found that the multi-task DNN could accurately segment the foreground, but its classification scores were influenced by background bias in all experiments where the bias was present. Therefore, the DNN's attention foci for the segmentation and classification tasks diverged and the ISNet significantly surpassed the model's resistance to background bias. **Here**, the multi-task U-Net could also precisely segmented the foreground in Stanford Dogs and COVID-19 detection, achieving 0.824 and 0.875 test intersection-over-union (IoU), respectively. However, its classification outputs were again influenced by synthetic bias (Table 2), allowing the model to be surpassed by Faster ISNets in all our tasks.
- **Standard attention mechanisms:** In [3], standard attention mechanisms (represented by the Vision Transformer and AG-Sononet), which do not learn from segmentation targets, could not reliably differentiate important foreground features from background bias. Thus, they considered the bias relevant and did not reduce shortcut learning, being surpassed by the ISNet in all tasks with background bias [3]. **Here**, the Vision Transformer and AG-Sononet paid attention to background bias in Stanford Dogs and in COVID-19 detection. The AG-Sononet is the network with the strongest focus on synthetic background bias in Stanford Dogs and in all experiments in [3]. As most attention mechanisms, its purpose is not to be resistant to spurious correlations in the background. Instead, the architecture can efficiently focus on the foreground when backgrounds represent clutter and noise [20].
- **RRR and ISNet Grad*Input:** In [3], we observed that, with respect to the optimization of input gradients (in RRR [24]) and gradient*input (in the ISNet Grad*Input), the optimization of LRP-ε (ISNet) better and more stably converged when using very deep backbones (DenseNet121) and large images (224×224), leading to superior resistance to background bias and o.o.d. accuracy [3]. The probable reason is that input gradients and Gradient*Input explanations are noisier than LRP-ε for deep neural networks, making convergence harder [3], [9]. **Here**, the optimization of input gradients (RRR) and Gradient*Input could only lead to high accuracy and robustness to background bias when DNNs were shallower, i.e., using a ResNet18 backbone. The methods were inaccurate and/or biased in all experiments using deeper backbones (ResNet50 or DenseNet121), both in this study and in [3].
- **GAIN:** In [3], we demonstrated that DNNs optimizing Grad-CAM (GAIN [21] and HAM [22]) could learn to produce deceiving Grad-CAM heatmaps, which showed no background attention for classifiers whose decisions were being influenced by background bias (spurious mapping) [3]. We also demonstrated that this phenomenon is inherent to the Grad-CAM formulation, and it cannot affect LRP optimization [3]. Accordingly, the ISNet surpassed GAIN in all experiments involving

background bias in [3]. **Here**, GAIN did not suffer spurious mapping in Stanford Dogs, but it did in COVID-19 classification and in a subset of Stanford Dogs in [3]. These results indicate that spurious mapping is a possible but not guaranteed outcome of Grad-CAM optimization, representing a shortcut solution to Grad-CAM-based losses. Dataset, model, and training dynamics can unpredictably define whether the DNN arrives at the biased solution or not, as mentioned in [3].

E. RUN-TIME AND TRAINING SPEED

The Faster ISNet and the original ISNet have the same run-time speed as a standard classifier because the creation of LRP heatmaps is not necessary after training; e.g., any ISNet with a DenseNet121 backbone will be as fast as a standard DenseNet121 at run-time. In COVID-19 detection, the segmentation-classification pipeline (U-Net followed by DenseNet121) was the second-best performing neural network, after the ISNet. Moreover, in the original ISNet study [3], it was the only benchmark model that could reliably avoid the influence of synthetic background bias on the classifier's decisions, like the ISNet. However, at run-time, the model is much slower than any ISNet variant, as it involves running two deep neural networks, instead of one. The original ISNet study compared the run-time ISNet with a DenseNet121 [10] backbone to the pipeline composed of a U-Net [16] and the DenseNet121. Considering mixed-precision, mini-batches of 10 images (of size $224 \times 224 \times 3$), and an NVIDIA RTX 3080, the ISNet was able to process an average of 353 samples per second, while the pipeline processed 207. Without mixed-precision, the ISNet processed 298 samples per second, and the pipeline 143. Thus, the ISNet was 70 to 108% faster [3]. Moreover, the model had only 8M parameters, while the pipeline had 39M, being about 5 times larger. Therefore, the original and the Faster ISNets are significantly faster and lighter than the pipeline after training. Such efficiency is especially important for its deployment in mobile or less capable devices.

The original ISNet's training time increases approximately linearly with the number of classes in the classification problem [3]. Considering a classification problem with 3 classes, 24183 training images and 2993 validation samples, the original ISNet took about 1300 s per epoch [3]. Meanwhile, the benchmark neural networks training times were: 1500 s for GAIN, 900 s for RRR, 320 s for the segmentation-classification pipeline (and 400 s per epoch when pre-training its U-Net), 240 s for the DenseNet121, 250 s for the multi-task U-Net, 160 seconds for the vision transformer, and 60 s for the AG-Sononet [3].

The Faster ISNet's training time does not increase linearly with the number of classes in the classification task. Figure 5 compares the training time for the original and the Faster ISNets, employing a log-log plot. We simulated 1 to 1000 classes in the classification task, by varying the number of output neurons in the networks. For realism, we set a fixed limit of GPU (graphics processing unit) memory: 10 gigabytes. Accordingly, the mini-batch size is always set to approximately utilize 10 gigabytes. The original ISNet memory utilization increases while we generate LRP heatmaps in parallel, thus we reduce batch size while increasing the number of classes. When batch size reaches one, we start generating heatmaps in series, which makes the model's training time increase linearly. In fig. 5, the original ISNet's batch size reaches one with about 10 classes (considering 10 GB of memory).

As expected, the original ISNet training time increases approximately linearly with the number of classes in the classification task. Meanwhile, the training time for all Faster ISNet variations (Selective, Dual, and Stochastic) is independent of the number of classes, like for the standard DenseNet121. The common classifier training time was 0.5 min per epoch, the Stochastic ISNet and the Selective ISNet achieved both 5 min per epoch, and the Dual ISNet, needing to produce two LRP heatmaps per image, required 13 minutes per epoch. With LRP Deep Supervision (LDS), the Stochastic ISNet, Selective ISNet, and Dual ISNet needed 9, 10 and 26 minutes per epoch, respectively. Considering the networks without LDS, the original ISNet was always slower than the Selective and Stochastic ISNets. It was faster than the Dual ISNet until about 3 classes in the classification problem, becoming increasingly slower afterwards. To put the numbers in fig. 5 into perspective, it would take about 16 hours to train the Selective or Stochastic ISNet for 200 epochs (considering the data and hardware we used to generate fig. 5). Meanwhile, the original ISNet would require 26 hours for 2 classes, 100 hours for 10 classes, and about 440 days with 1000 classes. Its training time could be reduced with more powerful hardware and parallelism over multiple GPUs. However, such settings represent higher cost and electrical energy consumption. Therefore, besides improving training time, the Faster ISNet expands the number of applications where LRP optimization can be feasibly implemented.

IV. LIMITATIONS AND FUTURE WORK

Here, we address limitations of this work. Computational cost may still be a concern [34], [35]. The Faster ISNet requires the creation of LRP heatmaps during training and the ISNet Loss requires specific hyper-parameter tuning. Figure 5 compares training time for the Faster ISNets and a standard classifier, and the original ISNet study [3] details a strategy for hyper-parameter tuning (explained in Appendix E). Here, we massively reduced training time in respect to the original ISNet in applications with many classes (Figure 5), and we introduced a new heuristic to accelerate hyper-parameter tuning (Section II-E).

The clinical application in this work is demonstrative of the Faster ISNet capacity of avoiding background bias attention. We do not perform clinical tests to assess real-world performance, and we listed the limitations of our X-ray datasets in Appendix G.

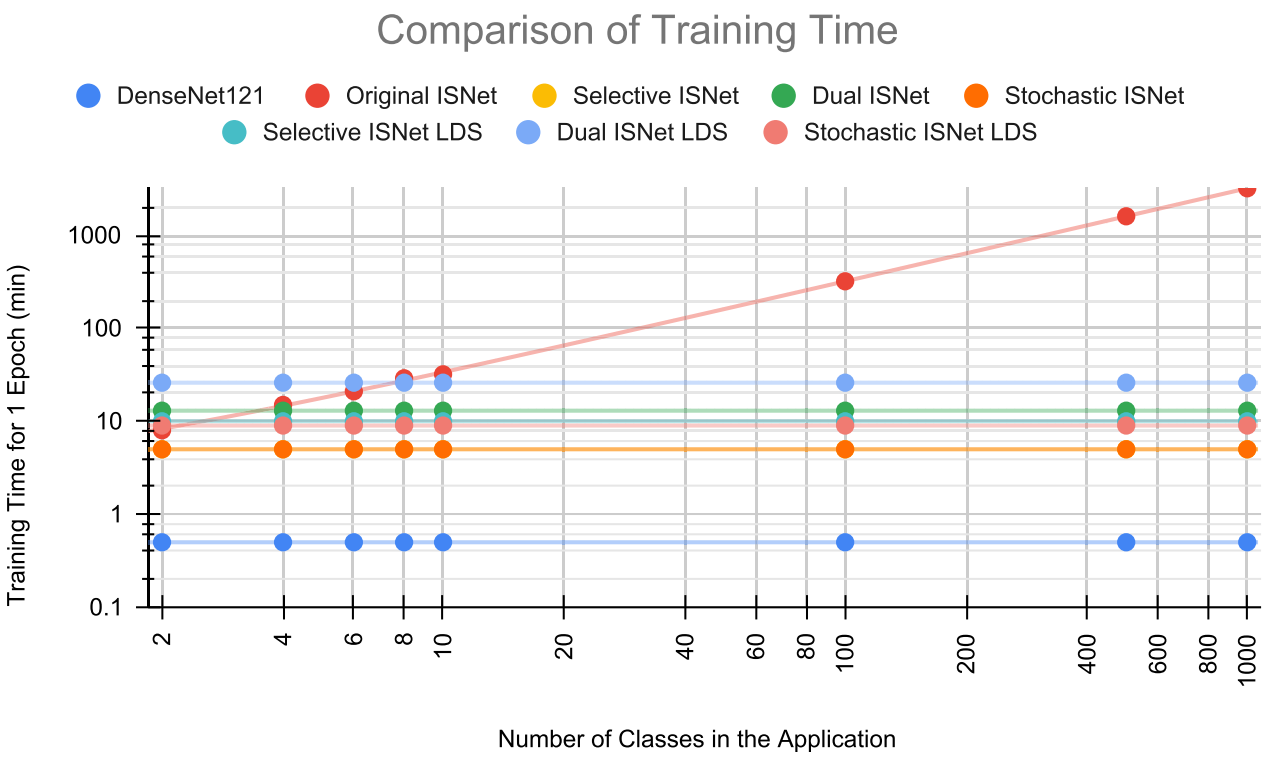

FIGURE 5. Training time analysis. The log-log plot displays the time for one epoch of 13932 224 × 224 × 3 images, and a varying number of categories in the classification problem. The networks ran in an NVIDIA RTX 3080, using 10GB of video memory, with mixed precision. The ISNet training time increases linearly with the number of classes, but the Faster ISNet training time is independent of this number. The Faster ISNet becomes significantly advantageous between 5 to 10 classes, and it trains 50× faster at 120 classes. At 1000 classes, the Faster ISNet can reduce training time from one year to one day.

Any ISNet requires LRP during training, increasing the overall complexity of the AI model. The original ISNet LRP implementation, the LRP Block, had about 2000 lines of code to implement 3 DNN architecture families (DenseNet, VGG and simple ''torch.Sequential'' networks). Here, we increased it to 4000 lines with the addition of the LRP-z^+ rule and the ResNet family. To convert new DNN architecture to the original ISNet, the LRP Block code needed to be manually updated. In this study, we alleviated this complexity limitation: our proposed LRP-Flex implementation, used in the Selective and Stochastic ISNets, requires 10× less code lines than the LRP Block, and it can produce LRP heatmaps for any ReLU-based DNN without any code update. We must note that no ISNet incurs more complexity at inference, because LRP is not needed at run-time.

The debate between post-hoc explanation methods like LRP, which interpret DNN decisions, and intrinsically interpretable architectures, such as some simpler machine learning techniques, is ongoing [36], [37]. A notable critique of saliency maps suggests that while they can identify image regions the neural networks found unimportant, they may not clearly explain how the networks used the relevant image parts [37]. This limitation is irrelevant to our work and the original ISNet study [3] because our focus is constraining networks to consider image backgrounds irrelevant, without addressing how they use foreground regions. Indeed, the ISNet heatmap loss is designed to minimally impact LRP foreground relevance, allowing the standard classification loss to guide how the network uses foreground features [3]. Another influential study argues that interpretability is a multi-faceted concept, and that post-hoc methods should set specific objectives and provide evidence that these objectives are satisfied by the proposed explanation techniques, rather than broadly claiming interpretability [36]. Here and in [3], we set a clear objective: to minimize the influence of background bias on classifiers. Our o.o.d. evaluation results on datasets with synthetic and natural background bias quantitatively demonstrate (Tables 1, 2 and 3) that minimizing backgrounds in LRP heatmaps satisfies this objective. We argue that ad-hoc explanations should be accompanied by comprehensive o.o.d. evaluation and not relied upon solely to determine AI trustworthiness.

We note that shortcut learning is not *only* caused by background bias [1]. Other forms of spurious correlations, such as colors in the foreground or object position may also cause shortcut learning and hinder generalization. Moreover, a generalization problem may also have causes beyond shortcut learning, such as label shifts between datasets [38]. Like the original ISNet and other explanation background minimization strategies, such as RRR [24] and GAIN [21], the Faster ISNet is solely designed to address the problem of background bias, not other causes of shortcut learning. The 7 applications in the original ISNet study delineate the model's use case: it succeeded in the 5 applications with

background bias but was not helpful in a dataset without background bias, nor in a case of label shift. The applicability of the ISNet beyond computer vision has not yet been tested. However, LRP works in other areas, such as tabular data or natural language processing (NLP) [14]. Therefore, we hypothesize that the ISNet could be expanded to other areas. For example, in NLP, one could choose words or sentences that should not influence the classifier's decisions, defining them as background, and use background relevance minimization with LRP (Faster ISNet) to minimize the DNN attention to these words and sentences. We leave the exploration of the Faster ISNet in areas beyond computer vision to future work.

V. CONCLUSION

This study introduced a simple, efficient and model-agnostic LRP implementation (LRP Flex), which explains arbitrary DNNs or converts them into ISNets; we improved the ISNet Loss and accelerated hyper-parameter tuning; we introduced LDS, improving LRP optimization convergence and accuracy; and we proposed the Faster ISNets. The original ISNet study elucidated the theoretical fundamentals of LRP optimization and justified why the ISNet surpasses multiple state-of-the-art benchmark DNNs, which include attention mechanisms and the optimization of Grad-CAM, input gradients and Gradient*Input [3]. Our empirical results further support the theoretical analyses in [3]. In both studies, the ISNets and the segmentation-classification pipeline were the only implemented DNNs that were consistently robust to background bias [3]. However, relying on 2 DNNs, the pipeline is much slower and heavier at inference; ISNets are about 2× faster and 5× smaller (with DenseNet121 backbone) [3]. ISNets add no run-time computational cost with respect to a standard classifier. Matching the original ISNet, the Faster ISNet was robust to background bias, hindered shortcut learning, and consistently achieved the best o.o.d. generalization in all experiments. Unlike the original ISNet, Faster ISNets' training time is independent of applications' number of classes. For 120 classes, they train about 50× faster than the ISNet; for 1000, they can reduce a 1-year training time to 1 day. By saving time, computational resources and energy, the Faster ISNet makes LRP optimization viable for a new plethora of applications.

ACKNOWLEDGMENT

The authors gratefully acknowledge the Data Science and Computation Facility Team for the support and assistance on the IIT High Performance Computing Infrastructure.

REFERENCES

[1] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Mach. Intell.*, vol. 2, no. 11, pp. 665–673, Nov. 2020.

[2] A. J. DeGrave, J. D. Janizek, and S.-I. Lee, "AI for radiographic COVID-19 detection selects shortcuts over signal," *Nature Mach. Intell.*, vol. 3, no. 7, pp. 610–619, May 2021.

[3] P. R. A. S. Bassi, S. S. J. Dertkigil, and A. Cavalli, "Improving deep neural network generalization and robustness to background bias via layer-wise relevance propagation optimization," *Nature Commun.*, vol. 15, no. 1, p. 126, Jan. 2024.

[4] A. Signoroni, M. Savardi, S. Benini, N. Adami, R. Leonardi, P. Gibellini, F. Vaccher, M. Ravanelli, A. Borghesi, R. Maroldi, and D. Farina, "BS-net: Learning COVID-19 pneumonia severity on a large chest X-ray dataset," *Med. Image Anal.*, vol. 71, Jul. 2021, Art. no. 102046.

[5] J. P. Cohen, P. Morrison, and L. Dao, "COVID-19 image data collection," 2020, *arXiv:2003.11597*.

[6] A. Shoeibi, M. Khodatars, M. Jafari, N. Ghassemi, D. Sadeghi, P. Moridian, A. Khadem, R. Alizadehsani, S. Hussain, A. Zare, Z. A. Sani, F. Khozeimeh, S. Nahavandi, U. R. Acharya, and J. M. Gorriz, "Automated detection and forecasting of COVID-19 using deep learning techniques: A review," *Neurocomputing*, vol. 577, Apr. 2024, Art. no. 127317.

[7] J. D. López-Cabrera, R. Orozco-Morales, J. A. Portal-Díaz, O. Lovelle-Enríquez, and M. Pérez-Díaz, "Current limitations to identify COVID-19 using artificial intelligence with chest X-ray imaging (Part II). The shortcut learning problem," *Health Technol.*, vol. 11, no. 6, pp. 1331–1345, Nov. 2021.

[8] G. Maguolo and L. Nanni, "A critic evaluation of methods for COVID-19 automatic detection from X-ray images," *Inf. Fusion*, vol. 76, pp. 1–7, Dec. 2021.

[9] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, no. 7, Jul. 2015, Art. no. e0130140.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[12] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[13] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *Proc. 1st Workshop Fine-Grained Vis. Categorization*, Jun. 2011, pp. 1–26.

[14] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: An overview," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Cham, Switzerland: Springer, 2019, pp. 193–209.

[15] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep Taylor decomposition," *Pattern Recognit.*, vol. 65, pp. 211–222, May 2017.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*. Cham, Switzerland: Springer, 2015, pp. 234–241.

[17] V. Birodkar, Z. Lu, S. Li, V. Rathod, and J. Huang, "The surprising impact of mask-head architecture on novel class segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 1–26.

[18] A. Kirillov, "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 4015–4026.

[19] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 7, pp. 1–26, 1997.

[20] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert, "Attention gated networks: Learning to leverage salient regions in medical images," 2018, *arXiv:1808.08114*.

[21] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," in *Proc. IEEE Conf. Comput. Vis.*, Jun. 2018, pp. 9215–9223.

[22] X. Ouyang, S. Karanam, Z. Wu, T. Chen, J. Huo, X. S. Zhou, Q. Wang, and J.-Z. Cheng, "Learning hierarchical attention for weakly-supervised chest X-ray abnormality localization and diagnosis," *IEEE Trans. Med. Imag.*, vol. 40, no. 10, pp. 2698–2710, Oct. 2021.

[23] A. Dosovitskiy, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–23.

[24] A. S. Ross, M. C. Hughes, and F. Doshi-Velez, "Right for the right reasons: Training differentiable models by constraining their explanations," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2662–2670, doi: 10.24963/IJCAI.2017/371.
[25] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," 2016, *arXiv:1605.01713*.
[26] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
[27] M. Ancona, E. Ceolini, C. Oztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," 2018, *arXiv:1711.06104*.
[28] S. Qiu, "Global weighted average pooling bridges pixel-level localization and image-level classification," 2018, *arXiv:1809.08264*.
[29] B. P. Welford, "Note on a method for calculating corrected sums of squares and products," *Technometrics*, vol. 4, no. 3, p. 419, Aug. 1962.
[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–24.
[31] D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems," *Mach. Learn.*, vol. 45, no. 2, pp. 171–186, Oct. 2001, doi: 10.1023/A:1010920819831.
[32] P. R. A. S. Bassi and R. Attux, "COVID-19 detection using chest X-rays: Is lung segmentation important for generalization?" *Res. Biomed. Eng.*, vol. 38, no. 4, pp. 1121–1139, Nov. 2022, doi: 10.1007/S42600-022-00242-Y.
[33] Y.-J. Jung, S.-H. Han, and H.-J. Choi, "Explaining CNN and RNN using selective layer-wise relevance propagation," *IEEE Access*, vol. 9, pp. 18670–18681, 2021.
[34] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.
[35] X. Bouthillier, P. Delaunay, M. Bronzi, A. Trofimov, B. Nichyporuk, J. Szeto, N. M. Sepahvand, E. Raff, K. Madan, and V. Voleti, "Accounting for variance in machine learning benchmarks," *Proc. Mach. Learn. Syst.*, vol. 3, pp. 747–769, Jun. 2021.
[36] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery," *Queue*, vol. 16, no. 3, pp. 31–57, Jun. 2018.
[37] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, May 2019.
[38] J. P. Cohen, M. Hashir, R. Brooks, and H. Bertrand, "On the limits of cross-domain generalization in automated X-ray prediction," in *Proc. 3rd Conf. Med. Imag. Deep Learn.*, vol. 121, 2020, pp. 136–155.

PEDRO R. A. S. BASSI received the bachelor's degree in electrical engineering and the master's degree in computer engineering from the State University of Campinas (UNICAMP), Brazil, in 2019 and 2021, respectively. He is currently pursuing the Ph.D. degree in data science and computation with the University of Bologna, Italy, and affiliated to the Istituto Italiano di Tecnologia (IIT). Since 2017, he has been conducting scientific research in the fields of machine learning and deep learning, encompassing applications in biomedical data analysis (e.g., X-ray, CT, and electroencephalography), computer vision, signal processing, and AI-assisted diagnosis. His main research interests include improving deep learning trustworthiness, out-of-distribution generalization, and explainability. He received the Certificate of Studies in Computer Engineering.

SERGIO DECHERCHI received the Laurea degree (summa cum laude) in electronic engineering and the Ph.D. degree in electronic engineering and computer science on machine learning and data mining from the University of Genoa, Italy, in 2007 and 2011, respectively. From 2011 to 2016, he was a Postdoctoral Researcher with the Department of Drug Discovery and Development (D3), Istituto Italiano di Tecnologia (IIT), Genoa, Italy, where he designed, developed and applied computational intelligence/chemistry methods to drug discovery. He is the Designer and the Developer of NanoShaper, a tool for molecular surface computation and pockets detection. In 2014, he co-founded BiKi Technologies s.r.l., a company dealing with molecular dynamics and machine learning methods for drug discovery. From 2017 to 2022, he was a Technologist with IIT. In 2022, he obtained the National Habilitation as an Associate Professor of computer science. Since 2023, he has been the Coordinator of the Data Science and Computation Facility, IIT. He is currently with the Department of Biophysics and Electronic Engineering (DIBE). He is the author of more than 70 papers in peer-reviewed journals and conferences in the fields of computational intelligence and computational chemistry. He has received some awards, EU/national/private grants, delivered several invited talks/lectures, and co-organized some workshops, such as ECAM/CECAM. He is a reviewer for EU and national funding agencies. He served as an Associate Editor for *Cognitive Computation* (Springer).

ANDREA CAVALLI received the Ph.D. degree in pharmaceutical sciences from the University of Bologna, in 1999. He did postdoctoral work at SISSA, Trieste, Italy; and ETH, Zürich, Switzerland. He was a Visiting Professor with the University of California at San Diego, in 2019. He is currently the Director of the European Center for Atomistic and Molecular Computations (CECAM), Ecole Polytechnique Fédérale de Lausanne, Lausanne. He is also a Professor of medicinal chemistry with the University of Bologna and leads the Computational and Chemical Biology Group, Istituto Italiano di Tecnologia, Genoa, Italy. He is also the Co-Founder of the high-tech start-up company BiKi Technologies. In particular, he has designed, developed, and applied new algorithms and codes to accelerate the discovery of new medicines in therapeutic areas, including cancer, neurodegenerative diseases, and neglected tropical diseases. He is the author of about 300 articles and has delivered over 120 invited lectures and seminars at international congresses and at prestigious international institutions. He is the co-inventor of 12 international PCT patents and patent applications. His research interests include computational chemistry and drug discovery. He is a member of the editorial board of numerous international scientific journals. From 2017 to 2022, he was the Chair of the international scientific organization QSAR, Chemoinformatic and Modeling Society (QCMS, former QSAR Society). He has been a reviewer and a committee member of several international funding agencies (EU, Austria, Switzerland, France, USA, Israel, Norway, and Portugal).

Open Access funding provided by 'Alma Mater Studiorum - Università di Bologna' within the CRUI CARE Agreement